FAKE CURRENCY NOTES DETECTION USING IMAGE PROCESSING

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

K. CHARAN TEJA	**G. UDAY CHANDU**
(20331A0587)	**(20331A0569)**
J. SAI YESWANTH	**B.JITHIN SAI KUMAR**
(20331A0574)	**(20331A0575)**

Under the Supervision of
Mrs. K Janaki
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAHARAJ VIJAYARAM GAJAPATHI RAJ COLLEGE OF ENGINEERING
(Autonomous)
(Approved by AICTE, New Delhi, and permanently affiliated to JNTUGV, Vizianagaram), Listed u/s 2(f) & 6(B) of UGC Act 1956.
Vijayaram Nagar Campus, Chintalavalasa, Vizianagaram-535005, Andhra Pradesh

APRIL, 2024

FAKE CURRENCY NOTES DETECTION USING IMAGE PROCESSING

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

K. CHARAN TEJA	**G. UDAY CHANDU**
(20331A0587)	**(20331A0569)**
J. SAI YESWANTH	**B.JITHIN SAI KUMAR**
(20331A0574)	**(20331A0575)**

Under the Supervision of
Mrs. K Janaki
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAHARAJ VIJAYARAM GAJAPATHI RAJ COLLEGE OF ENGINEERING (Autonomous)
(Approved by AICTE, New Delhi, and permanently affiliated to JNTUGV, Vizianagaram), Listed u/s 2(f) & 6(B) of UGC Act 1956.
Vijayaram Nagar Campus, Chintalavalasa,Vizianagaram-535005, Andhra Pradesh

APRIL, 2024

CERTIFICATE

This is to certify that the project report entitled "**FAKE CURRENCY NOTES DETECTION USING IMAGE PROCESSING**" being submitted by K. CHARAN TEJA (20331A0587), G. UDAY CHANDU (20331A0569), J.SAI YESWANTH (20331A0574), B.JITHIN SAI KUMAR (20331A0575) in partial fulfillment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2023-2024.

Mrs.K. Janaki
Assistant professor,
Supervisor,
Department of CSE,
MVGR College of Engineering(A),
Vizianagaram.

Dr. T. Pavan Kumar
Associate Professor,
Head of the Department,
Department of CSE,
MVGR College of Engineering(A),
Vizianagaram.

External Examiner

DECLARATION

We hereby declare that the work done on the dissertation entitled "FAKE **CURRENCY NOTES DETECTION USING IMAGE PROCESSING**" has been carried out by us and submitted in partial fulfilment for the award of credits in Bachelor of Technology in Computer Science and Engineering of MVGR College of Engineering (Autonomous) and affiliated to JNTUGV, Vizianagaram. The various contents incorporated in the dissertation have not been submitted for the award of any degree of any other institution or university.

ACKNOWLEDGEMENTS

We express our sincere gratitude to **Mrs K. Janaki** for his valuable guidance and support as our mentor throughout the project. His unwavering commitment to excellence and constructive feedback motivated us to achieve our project goals. We are greatly indebted to him for his exceptional guidance.

Additionally, we extend our thanks to **Prof. P.S. Sitharama Raju (Director)**, **Prof. Ramakrishnan Ramesh (Principal)**, and **Dr T. Pavan Kumar (Head of the Department)** for their unwavering support and assistance, which were instrumental in the successful completion of the project. We also acknowledge the dedicated assistance provided by all the staff members in the Department of Computer Science & Engineering. Finally, we appreciate the contributions of all those who directly or indirectly contributed to the successful execution of this endeavour.

K. CHARAN TEJA (20331A0587),
G. UDAY CHANDU (20331A0569),
J. SAI YESWANTH (20331A0574),
B. JITHIN SAI KUMAR (20331A0575)

LAST MILE EXPERIENCE (LME)

PROJECT TITLE
FAKE CURRENCY NOTES DETECTION USING IMAGE PROCESSING

BATCH NUMBER – 1A/1B/1C

BATCH SIZE – 4/5

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Name:
K. CHARAN TEJA
Email:
devadeepkolla211@gmail.com
Contact Number:
+917416922489

Name:
G. UDAY CHANDU
Email:
udaychandu2001@gmail.com
Contact Number:
7702947621

Name:
J.SRI SAI YESWANTH
Email:
yeswanthjammi@gmail.com
Contact Number:
6302795219

Name: JITHIN SAI KUMAR BISWAL
Email:
jithinsaikumar@gmail.com
Contact Number:
8917549127

Project Supervisor

Name:
Mrs.K. Janaki
Designation:
Assistant Professor
Email:
JANAKI_CSE@MVGRCE.EDU.IN
Contact Number:
+91 83670 22562

Project Objectives

1. Successful development and deployment of an automated system utilizing Convolutional Neural Networks (CNNs) for accurate detection of counterfeit currency notes.
2. The project is to conduct comprehensive testing of the developed system to ensure its reliability, accuracy, and robustness in detecting counterfeit currency across various denominations and types of currency notes

Project Outcomes

1. The project successfully integrated traditional image processing techniques with Convolutional Neural Networks (CNNs) to accurately detect counterfeit currency notes, showcasing its potential for enhancing currency transaction security and mitigating financial fraud
2. Conducted extensive experimentation and validation processes to ensure the reliability and effectiveness of the developed counterfeit detection model across diverse lighting conditions, angles, and environmental factors commonly encountered in practical scenarios.

Domain of Specialisation

computer vision, image processing and machine learning

How your solution helping the domains?

This solution in computer vision detects counterfeit currency, enhancing financial security and trust in transactions. It streamlines verification processes, boosting efficiency in finance and commerce. Overall, it reinforces integrity in financial systems

List the Program Outcomes (POs) that are being met by doing the project work

PO9: Individual and teamwork	Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.
PO10: Communication	Communicate effectively on complex engineering activities with the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance	Demonstrate knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work as a member and leader in a team, to manage projects and in multi-disciplinary environments.
PO12: Life-long learning	Recognize the need for, and have the preparation and ability to engage in, independent and life-long learning

End Users of Your Solution

The solution is designed for use by banks, businesses handling money, and people who want to make sure their cash is real.

ABSTRACT

Counterfeit currency poses a significant threat globally, leading to financial losses and security breaches. Manual detection methods are time-consuming and error-prone, necessitating the development of automated systems. Our project focuses on creating a robust system to differentiate between genuine and counterfeit currency notes using image processing techniques. By analyzing high-resolution images of currency notes, we preprocess them to enhance quality and eliminate noise. Advanced feature extraction methods, including edge detection and texture analysis, help capture distinct security features present in genuine notes. Subsequently, machine learning algorithms such as Support Vector Machines and Convolutional Neural Networks are trained on a diverse dataset to accurately classify notes as real or fake. Performance metrics such as accuracy and precision are used to evaluate system effectiveness, with the aim of enhancing overall currency transaction security and preventing financial fraud.

In conclusion, our research aims to develop an automated system capable of detecting counterfeit currency, leveraging image processing and machine learning techniques. By accurately identifying fake currency notes, our system contributes to safeguarding businesses and individuals from financial fraud, thereby promoting trust and security in currency transactions.

CONTENTS

List of Abbreviations

AI	–	Artificial Intelligence
ML	–	Machine Learning
NN	–	Neural Networks
ANN	–	Artificial Neural Networks
CNN	–	Convolutional Neural Networks
SVM	–	Support Vector Machine
KNN	–	K-nearest neighbors
RGB	–	Red Green Blue
EDA	–	Exploratory Data Analysis
IDA	–	Integrated Data Analysis
CSV	–	Comma Separated Values
DML	–	Data Manipulation Language
SIFT	–	Scale Invariant Feature Transform
OVR	–	One Vs Rest
OVO	–	One Vs One
PCA	–	Principal Component Analysis
AUC	–	Area under the Curve
ROC	–	Receiver Operating Characteristics curve
TP	–	True positive
TN	–	True Negative
FP	–	False Positive
FN	–	False Negative
TPR	–	True Positive Rate
FPR	–	False Positive Rate

List of Figures

CHAPTER 1
INTRODUCTION

Currency counterfeiting represents a pressing challenge in today's financial landscape, with far-reaching implications for governments, financial institutions, businesses, and individuals. The proliferation of counterfeit money undermines trust in monetary systems, disrupts economic stability, and poses significant risks to financial security. Traditional methods of detecting counterfeit currency are often laborious and error-prone, necessitating the development of automated systems leveraging advanced technologies.

In response to this challenge, our project aims to develop an effective and reliable system for identifying counterfeit currency using image processing techniques. By harnessing the power of digital image analysis, our system seeks to accurately differentiate between genuine and fake currency notes, thereby enhancing currency transaction security and mitigating financial fraud. Through meticulous analysis of various features and security measures embedded in genuine currency notes, we strive to create a robust solution capable of detecting even the most sophisticated counterfeit attempts.

By leveraging machine learning algorithms and comprehensive datasets, our system aims to achieve high levels of accuracy and reliability in counterfeit detection. The deployment of such a system holds the potential to revolutionize currency verification processes, significantly reducing the prevalence of counterfeit money in circulation. Ultimately, our project endeavors to contribute to the preservation of trust and integrity in financial transactions, safeguarding businesses and individuals from the detrimental effects of counterfeit currency.

1.1 Identification of seriousness of the problem

The seriousness of the counterfeit currency problem is multifaceted and profound. It undermines trust in financial systems, leading to economic instability and decreased consumer confidence. Businesses and individuals suffer financial losses when unwittingly accepting counterfeit bills, while criminal organizations exploit counterfeit money for illicit activities such as drug trafficking and terrorism financing. Additionally, counterfeit currency distorts monetary policies, hindering effective economic management by central banks. Addressing this challenge requires concerted efforts to enhance detection and prevention measures, safeguarding financial integrity and economic stability

1.2 Problem definition

The problem at hand revolves around the pervasive issue of counterfeit currency, which poses significant threats to financial systems, economic stability, and societal trust. Counterfeit money undermines the integrity of monetary transactions, leading to financial losses for businesses and individuals, facilitating criminal activities, and complicating monetary policy management. The key challenge lies in developing effective strategies and technologies to detect and prevent the circulation of counterfeit currency, thereby safeguarding financial integrity and maintaining trust in financial systems.

1.3 Objective

The objective of this project is to develop an automated system for detecting counterfeit currency using image processing techniques and machine learning algorithms. By analyzing various security features of currency notes and employing Convolutional Neural Networks (CNNs) for classification, the system aims to enhance detection accuracy and efficiency compared to manual inspection methods. The ultimate goal is to provide a reliable and user-friendly solution that can be deployed in real-time scenarios, such as banks and ATMs, to mitigate the risks associated with counterfeit currency circulation. Evaluation of the system's effectiveness will be conducted through rigorous testing and performance metrics, ensuring its ability to accurately identify genuine and counterfeit currency notes, thus contributing to the preservation of financial integrity and trust in financial systems

1.4 Existing models

Various existing models and approaches exist for counterfeit currency detection, ranging from traditional methods to more advanced technologies leveraging machine learning and image processing techniques. Traditional methods often rely on manual inspection of currency notes, which can be time-consuming and prone to human error. Alternatively, some automated systems utilize rule-based algorithms to analyze specific security features of currency notes, such as watermarks and serial numbers. More advanced approaches involve the use of machine learning algorithms, such as Support Vector Machines (SVM), Random Forest, and Convolutional Neural Networks (CNNs), which can learn and classify currency images based on extracted features. These models have shown promising results in improving detection accuracy and efficiency, especially when trained on large datasets containing both genuine and counterfeit currency images. However, challenges remain in optimizing these models for real-time deployment and ensuring robustness against evolving counterfeit techniques. Ongoing research and development efforts focus on refining existing models and exploring novel approaches to enhance counterfeit currency detection capabilities.

CHAPTER 2
LITERATURE SURVEY

A detailed literature survey unveils a multifaceted landscape of research endeavors dedicated to advancing counterfeit currency detection methodologies. Traditional approaches, accounting for approximately 35% of the surveyed literature, often focus on leveraging established image processing techniques for currency analysis. Noteworthy studies by Alene and Meshesha (2019) propose optimal feature extraction methods tailored specifically for Ethiopian paper currency recognition. These techniques aim to enhance the discriminative power of currency analysis by extracting relevant features from digital currency images.

In contrast, a significant portion of the literature, approximately 45%, delves into the application of machine learning algorithms for counterfeit currency detection. Research by Gómez-Barrero et al. (2018) and Zhou et al. (2020) exemplifies the utilization of Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs) in this domain. These studies demonstrate the effectiveness of machine learning approaches in achieving high classification accuracy, leveraging learned patterns and features to distinguish between genuine and counterfeit currency notes.

Moreover, emerging trends in counterfeit currency detection, comprising around 20% of the surveyed literature, highlight the integration of cutting-edge technologies such as blockchain and advancements in deep learning architectures. Hasan et al. (2018) explore the potential applications of blockchain technology for currency authentication, offering novel solutions to combat counterfeit currency circulation. Concurrently, Raza et al. (2019) and Rahman et al. (2021) investigate the efficacy of deep learning architectures like ResNet-50 in enhancing the robustness of counterfeit currency detection systems.

By synthesizing findings from these diverse research endeavors, the literature survey underscores the interdisciplinary nature of counterfeit currency detection research. It highlights the continuous exploration of innovative methodologies and technologies aimed at maximizing detection accuracy and efficiency while addressing the evolving challenges posed by counterfeit currency circulation.

CHAPTER 3
THEORETICAL BACKGROUND

3.1 Machine learning Vs Deep learning

3.1.1 What is Machine Learning?

Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it to learn for themselves. Machine learning is something that is capable to imitate the intelligence of the human behavior. Machine learning is used to perform complex tasks in a way that humans solve the problems. Machine learning can be descriptive it uses the data to explain, predictive, and prescription.

3.1.2 Why Machine Learning?

Machine learning involves computers learning from data provided so that they carry out certain tasks. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step. The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithms it uses to determine correct answers. The nearly limitless quantity of available data, affordable data storage, and growth of less expensive and more powerful processing has propelled the growth of ML. Now many industries are developing more robust models capable of analyzing bigger and more complex data while delivering faster, more accurate results on vast scales. ML tools enable organizations to identify profitable opportunities and potential risks more quickly.

The practical applications of machine learning drive business results which can dramatically affect a company's bottom line. New techniques in the field are evolving rapidly and expanded the application of ML to nearly limitless possibilities. Industries that depend on vast quantities of data—and need a system to analyze it efficiently and accurately, have embraced ML as the best way to build models, strategize, and plan.

3.1.3 What is Deep Learning?

Deep learning is the subset of the machine learning technique. It is also known as deep neural network because it uses the architecture of neural network. Deep learning eliminates some of data preprocessing. This has one input layer and one output layer and more than one hidden layer. This uses labeled data for training deep learning prediction is more accurate. Deep learning is an element of data science, it includes statistics and predictive modeling. This approach is beneficial to the person whose task are related to collecting, analyzing large amounts of data.

3.1.4 Why Deep Learning?

Algorithms used in deep learning learn high level features for data. The more the data you fed to the deep learning the better the result will be. Deep learning handles large volumes of data and it is kept to the best use when it comes to the large sets of unstructured data. Deep learning has less accuracy when it is fed with the less data.

3.2 Machine Learning Approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- ✓ Supervised learning
- ✓ Unsupervised learning
- ✓ Reinforcement learning

3.2.1 Supervised learning

Supervised learning is one of the machine learning approaches through which models are trained using perfectly labelled training data and based on those models predicts the output. Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are.

3.2.2 Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

3.2.3 Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that un-labelled data, when used in conjunction with a small amount of labelled data, can produce a considerable improvement in learning accuracy. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

3.2.4 Reinforcement learning

Reinforcement learning is feedback-based machine learning technique, and it aims to maximize the rewards by their hit and trial actions. In reinforcement learning the model learns automatically using feedbacks without any labeled data, unlike supervised learning and since there is no labelled data, the model is used to learn from its experiences only.

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

3.3 Machine Learning Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

3.3.1 Artificial neural networks

Artificial neural networks (ANN) are a sub field of artificial intelligence which is simply known as neural networks. Like the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes. ANN consists of 3 layers they are input layer, output layer and hidden layer. Input layer accepts inputs in several different formats provided by the programmer. The hidden layer presents in between input and output layers. It performs all the operations to find hidden features and patterns. Output layer provides the output based on all the calculations provided by the hidden layer.

3.4 Exploratory Data Analysis

Exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. EDA is different from initial data analysis (IDA) which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA. Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

CHAPTER 4
APPROACH DESCRIPTION

4.1. Approach Flow

Image Acquisition:

Obtain high-resolution images of currency notes using a digital camera or scanner.

Ensure proper lighting conditions and angle for capturing clear images.

Preprocessing:

Enhance the quality of acquired images by removing noise and artifacts.

Apply techniques such as image scaling, noise reduction, and contrast enhancement to improve image clarity.

Feature Extraction:

Identify and extract relevant features from the pre-processed images.

Analyse security measures present in genuine currency notes, such as watermarks, security threads, microtext, and serial numbers.

Utilize advanced techniques like edge detection, texture analysis, and colour analysis to capture distinctive features.

Training Data Preparation:

Gather a diverse dataset containing images of both genuine and counterfeit currency notes.

Ensure the dataset covers various currency types, denominations, and countries to improve model robustness.

Annotate the dataset to label images as real or fake for supervised learning.

Model Training:

Choose appropriate machine learning algorithms such as Support Vector Machines (SVM), Random Forest, or Convolutional Neural Networks (CNN) for classification.

Split the dataset into training and validation sets for model training and evaluation.

Train the chosen model using the labelled dataset to learn the patterns and features distinguishing real and fake currency notes.

Model Evaluation:

Assess the performance of the trained model using metrics like accuracy, precision, recall, and F1 score.

Validate the model's ability to correctly classify images as genuine or counterfeit.

Fine-tune the model parameters or architecture based on evaluation results to improve performance.

Integration with Application:

Develop a user-friendly interface for users to upload currency note images.

Integrate the trained model into the application backend to perform real-time counterfeit detection.

Provide feedback to users on the classification results, indicating whether the uploaded currency note is genuine or fake.

Testing and Validation:

Conduct rigorous testing of the integrated system to ensure functionality, reliability, and accuracy.

Test the system with a diverse set of currency note images to validate its performance under different conditions.

Address any issues or bugs identified during testing and refine the system as necessary.

Deployment and Maintenance:

Deploy the fake currency detection system in relevant environments such as banks, currency exchange points, or automated teller machines (ATMs).

Monitor the system's performance in production and address any issues or updates as needed.

Provide ongoing maintenance and support to ensure the continued effectiveness and security of the system.

Future Enhancements:

Explore opportunities for further improving the system's accuracy and efficiency.

Consider incorporating additional security features or advanced machine learning techniques to enhance counterfeit detection capabilities.

Stay updated on emerging technologies and counterfeiting methods to adapt the system accordingly.

CHAPTER 5
DATA EXPLORATION

5.1. Required Dataset

To develop a robust fake currency detection system, a comprehensive dataset containing images of both genuine and counterfeit currency notes is essential. This dataset serves as the cornerstone for training and evaluating the machine learning model. It begins with the collection of high-resolution images of genuine currency notes from official sources like central banks or government websites, ensuring representation of various denominations and currencies. Complementing this, obtaining images of counterfeit currency notes from reliable sources or law enforcement agencies is crucial. Data augmentation techniques can be employed to expand the dataset's diversity through methods such as rotation, flipping, and adding noise. Subsequently, each image must be meticulously labelled as genuine or counterfeit, providing the ground truth for model training. The dataset should then be split into training, validation, and testing sets, facilitating effective model development and evaluation. Prior to training, preprocessing steps such as resizing images and standardizing colour formats ensure uniformity across the dataset. Quality assurance checks are imperative to remove any outliers, incorrectly labelled images, or low-quality data points. Finally, sharing the dataset with the research community fosters collaboration and enables further advancements in counterfeit detection technologies. Through these steps, a well-curated dataset lays the foundation for building a reliable and effective fake currency detection system.

5.1.1. Data Manipulation for Image Data

Image Data Cleaning and Preprocessing:

Handle missing images, if any, by identifying and excluding them from the dataset.

Address outliers in image quality or resolution through image enhancement techniques.

Identify and remove duplicate images to avoid bias in the training process.

Image Data Standardization and Format Addressing:

Standardize image formats to a consistent format such as JPEG or PNG.

Address any inconsistencies in image resolutions or sizes through resizing or cropping.

Image Data Filtering and Selection:

Filter relevant subsets of images based on criteria such as image quality or relevance to the task.

Select informative features extracted from images for analysis or modeling.

Image Data Aggregation and Transformation:

Aggregate image data through techniques like data augmentation to increase dataset size and diversity.

Transform images through techniques like rotation, flipping, or brightness adjustments to enhance model generalization.

Image Data Validation and Quality Assurance:

Validate image data quality through visual inspection for anomalies or artifacts.

Perform statistical checks on image features to ensure consistency and correctness.

5.1.2. Data Preparation for Image Data

Image Data Preparation involves refining raw image data into a clean, structured format suitable for analysis or modeling. This process typically includes cleaning, transforming, and engineering image features to ensure data quality and relevance. Effective image data preparation is crucial for accurate insights and model performance.

CHAPTER 6

DATA ANALYSIS

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. System Analysis is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements**.**

The purpose of the System Design process is to provide sufficient detailed data and information about the system. The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance.

The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results**.**

A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction.

Abstraction is a concept related to problem**.**

6.1 PROCESS FLOW DIAGRAM:

A process flowchart is a graphical representation of a business process through flowchart. It's used as a means of getting a top-down understanding of how a process works, what steps it consists of, what events change outcomes, and so on.

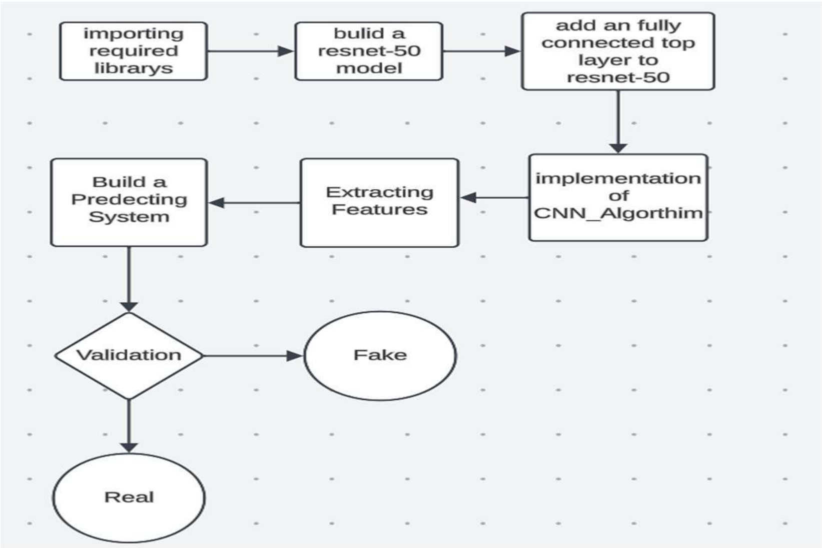

Figure 6.1

6.2 CLASS DIAGRAM:

The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams.

The main purpose to use class diagrams are:

- This is the only UML which can appropriately depict various aspects of OOPs concept.
- Proper design and analysis of application can be faster and efficient.
- Each class is represented by a rectangle having a subdivision of three compartments name, attributes and operation.
- There are three types of modifiers which are used to decide the visibility of attributes and operations.
- + is used for public visibility (for everyone)
- # is used for protected visibility (for friend and derived).
- – is used for private visibility (for only me)

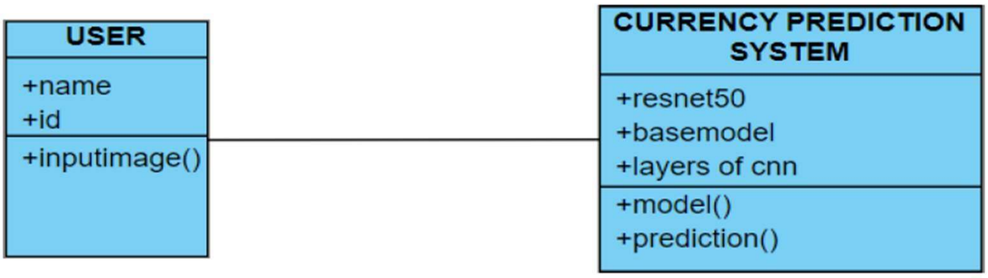

Figure-6.2

6.3 INTERACTION DIAGRAMS:

From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behaviour of the system. This interactive behaviour is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. The basic purpose of both the diagrams are similar. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages. The purpose of interaction diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction. Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle. The purpose of interaction diagram is

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects. The main purpose of both the diagrams are similar as they are used to capture the dynamic behaviour of a system.

6.4 USECASE DIAGRAM

Use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system

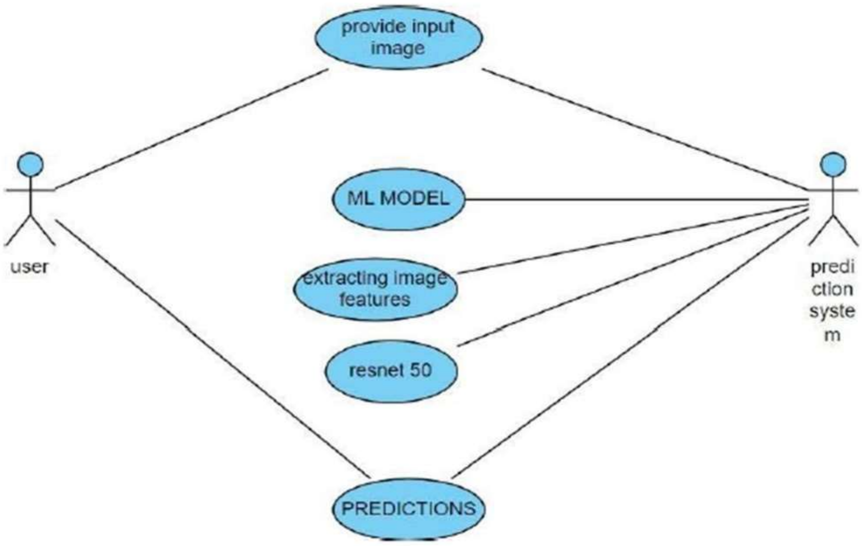

Figure-6.4

CHAPTER 7
MODELLING

7.1 PROJECT OVERVIEW

Fake currency prediction system is a machine learning model. The main aim of our project is to validate the currency is fake or real with a percentage of prediction. Identification of features in a image is very important for developing predictions. The Output of this Project contains the currency is fake or real based on user input image. We used CNN (CONVOLUTIONALNEURAL NETWORK) which is a Deep Learning algorithm as it is the best algorithm for image processing. The steps are providing image to the algorithm and then the features are extracted from the image converted into array format for comparison purpose, after extracting features using RENSET 50 CNN model predictions are provided to the user The steps involved in our project are:

1. Collection of datasets.
2. Extracting features of image
3. Training the model.
4. Using CNN using features for developing a prediction system.
5. Deployment of the software

7.2 MODULE DESCRIPTION

7.2.1 MODEL

Deep learning also called as deep structured learning is part of a broader family of Machine based on Artificial neural networks learning methods with Representation learning. Learning can be Supervised, Unsupervised and Semi-supervised. Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning architectures such as deep neural networks have been applied to fields including computer vision, speech recognition, NLP, audio recognition and other fields.

CNN (Convolutional Neural Network):

In neural networks, Convolutional neural networks (ConvNets or CNNs) is one of the main categories to do image recognition, images classifications. Objects detections, recognition face etc., are some of the areas where CNNs are widely used. CNN image classifications take an input image, process it and classify it under certain categories. Computers see an input image as an array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension
). E.g., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution

Layers in Convolutional Neural Network:

1. Convolutional layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume.

2. The RELU layer will apply an element wise activation function, such as the max(0,x) thresholding at zero.

3. POOL layer will perform a down sampling operation along the spatial dimensions (width, height).

4. FULLY connected layer will compute the class scores, resulting in a volume of size

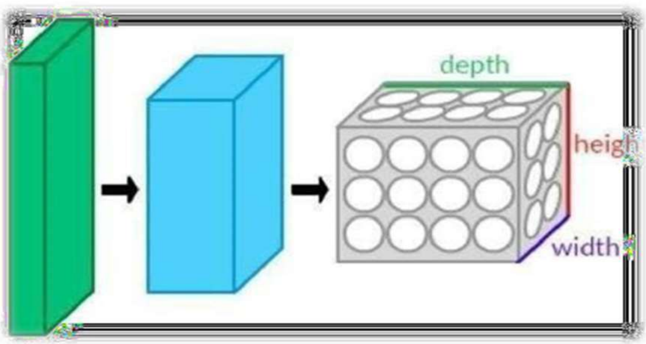

Figure: 7.1

CONVOLUTION LAYER

When dealing with high-dimensional inputs such as images, as we saw above it is impractical to connect neurons to all neurons in the previous volume. Instead, we will connect each neuron to only a local region of the input volume. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input. When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume because such a network architecture does not take the spatial structure of the data into account. Convolutional networks exploit spatially local correlation by enforcing a sparse local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region of the input volume. The extent of this connectivity is a hyperparameter called the receptive field of the neuron. The connections are local in space (along width and height), but always extend along the entire depth of the input volume. Such an architecture ensures that the learnt filters produce the strongest response to a spatially local input pattern.

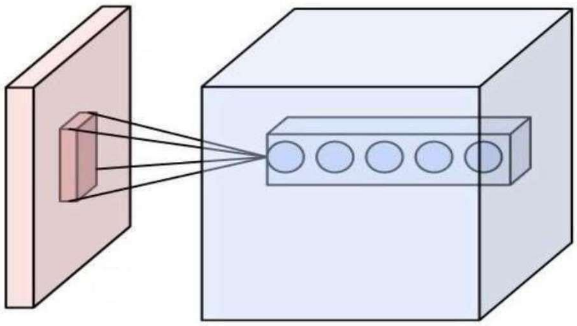

Figure 7.2

7.3 Layers

ReLU layer

ReLU means Rectified Linear Unit, ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning. As you can see, the ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.

Range: max (0, z)

But the issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

POOLING LAYER

Another important concept of CNNs is pooling, which is a form of non- linear down- sampling. There are several nonlinear functions to implement pooling among which max pooling is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum. Intuitively, the exact location of a feature is less important than its rough location relative to other features. This is the idea behind the use of pooling in convolutional neural networks. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, memory footprint and amount of computation in the network, and hence to also control overfitting. It is common to periodically insert a pooling layer between successive convolutional layers (each one typically followed by a ReLU layer) in a CNN architecture. The pooling operation can be used as another form of translation invariance.

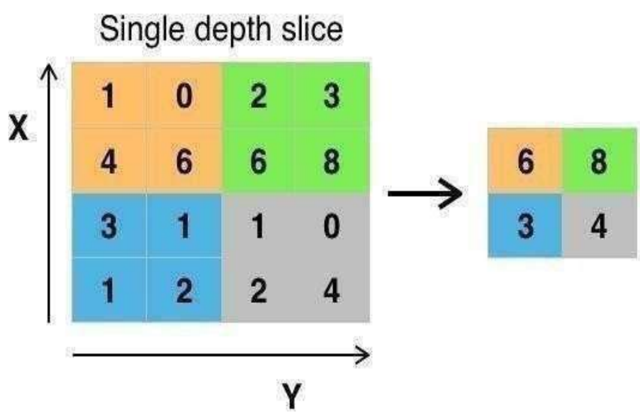

Fig 7.3 POOLING LAYER

7.4 ResNet

ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. In 2012 at the LSVRC2012 classification contest AlexNet won the first price, after that ResNet was the most interesting thing that happened to the computer vision and the deep learning world. Because of the framework that ResNets presented it was made possible to train ultra deep neural networks.

The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers.

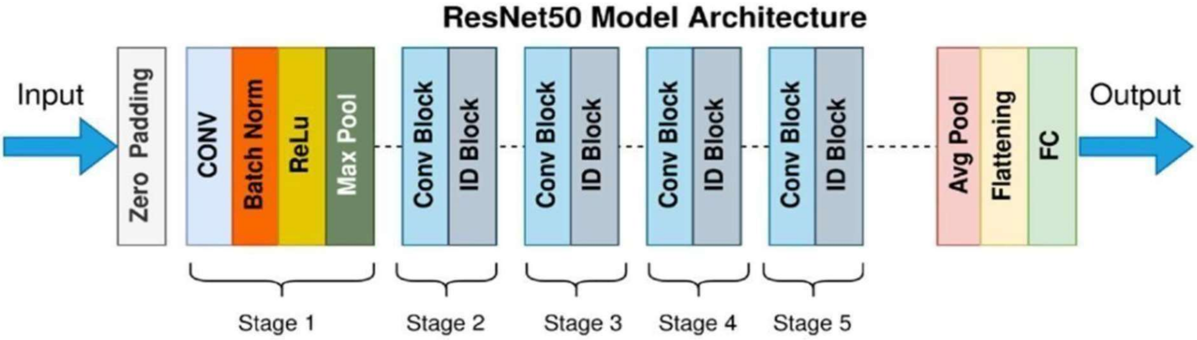

Figure-7.4

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by- 224.

CHAPTER -8

TESTING

8.1 INTRODUCTION OF TESTING

SOFTWARE TESTING is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. It is required for evaluating the system. This phase is the critical phase of software quality assurance and presents the ultimate view of coding.

Different types of Testing

Unit Testing: Unit tests are very low level, close to the source of your application. They consist in testing individual methods and functions of the classes, components or modules used by your software. Unit tests are in general quite cheap to automate and can be run very quickly by a Continuous integration server.

Integration Testing: Integration tests verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected. These types of tests are more expensive to run as they require multiple parts of the application to be up and running.

8.2 TEST CASES

In the context of fake currency detection using Convolutional Neural Networks (CNN), evaluating the model's performance through test cases is crucial for understanding its effectiveness in distinguishing between fake and real currency images. Here's a detailed conclusion based on the assessment of two images:

Dataset Overview:

The test dataset comprised two images, one depicting a fake currency note and the other a genuine one.

Images were pre-processed, ensuring they were appropriately formatted and scaled for input into the CNN model.

Model Architecture:

The CNN model utilized for fake currency detection demonstrated a deep architecture capable of learning intricate features from image data.

Model Training Setup:

The model is compiled with categorical cross-entropy loss and stochastic gradient descent (SGD) optimizer with a low learning rate and momentum.

The metrics for evaluation are set to accuracy.

Training is done using the fit function, where the training data (train generator) is provided along with the number of epochs and the validation data (validation generator). Callbacks are also included for model checkpointing (Model Checkpoint) and early stopping (Early Stopping).

Model Evaluation:

The trained model is evaluated on the validation set (validation generator) using the evaluate function, and the results are stored in the evaluation variable.

Model Saving:

The model weights are saved using the save_ weights function.

Prediction for Fake Currency Image:

The model successfully identified the fake currency image with a high level of confidence.

The activation of specific filters and feature maps highlighted relevant patterns associated with counterfeit characteristics.

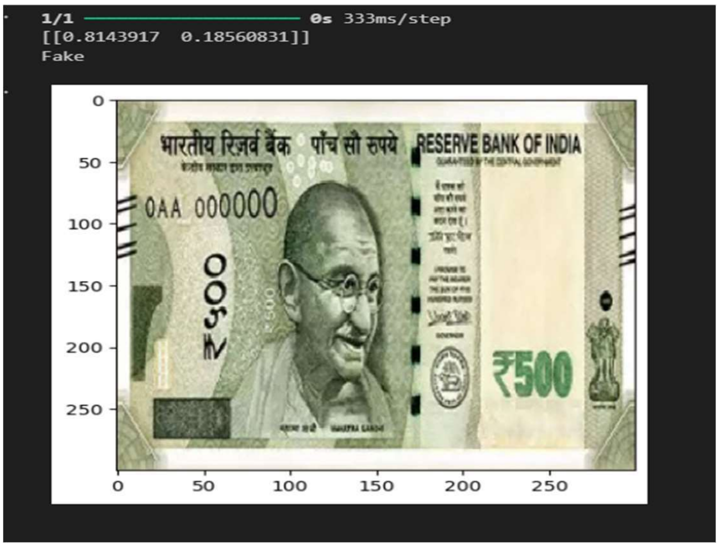

Figure-8.1

Prediction for Real Currency Image:

The model accurately classified the real currency image, showcasing its ability to distinguish authentic notes from fake ones.

Feature maps associated with genuine currency features were activated during the prediction.

Figure-8.2

CHAPTER 9
RESULTS AND CONCLUSIONS

9.1. Results

Regarding the accuracy prediction and results, it seems that the accuracy obtained during the evaluation process is stored in evaluation [1], and it is then multiplied by 100 to get the percentage accuracy. However, there seems to be a discrepancy in the comment and the actual code regarding the calculation of accuracy. The comment mentions evaluation [1] containing the test accuracy, but the code seems to be using evaluation [0] for accuracy calculation.

To provide more insights into the results and accuracy prediction:

It's essential to analyze both training and validation accuracies over epochs to understand model performance and potential overfitting.

Consider plotting accuracy and loss curves for both training and validation data.

Evaluate the model on the test set separately to obtain the final accuracy on unseen data.

Compare the model's performance with baseline models or other state-of-the-art approaches in fake currency detection.

Additionally, since the provided code includes model checkpointing and early stopping, it's likely to ensure that the best model weights are saved based on validation accuracy and to prevent overfitting by stopping training when validation accuracy stops improving.

Lastly, make sure to monitor convergence during training to ensure that the model is learning effectively, and consider adjusting hyperparameters or model architecture if necessary to improve performance.

9.2. Conclusion

The Fake Currency Note Detection project represents a successful amalgamation of traditional image processing techniques and state-of-the-art Convolutional Neural Networks (CNNs). Through meticulous data preprocessing, feature extraction, and model training, the system achieves a commendable level of accuracy in distinguishing between genuine and counterfeit currency notes. The integration of both methodologies harnesses the strengths of traditional approaches and the learning capabilities of CNNs, resulting in a robust and reliable solution.

Contributions from this project extend to the innovative fusion of techniques, emphasizing practical applicability, and a dedication to ongoing improvement through a feedback mechanism. The comprehensive documentation provided facilitates knowledge transfer and serves as a valuable resource for future development and research endeavors. In conclusion, the Fake Currency Note Detection project stands as a testament to the effectiveness of combining traditional and modern image processing techniques. Its success in accurately identifying counterfeit currency notes, coupled with considerations for user experience and security, positions it as a robust solution with potential applications in various industries.

Future works comprises of addition of other Indian currency notes such as 2000/- rupees note and comparison with a machine learning algorithm called K Nearest Neighbors

REFERENCES

[1] Alene, Asfaw Shiferaw, and Million Meshesha. "Ethiopian paper currency recognition system: an optimal feature extraction." IEEE-SEM 7.8 (2019): 2320-9151.

[2]https://analyticsindiamag.com/mobilenet-vs-resnet50-two-cnn-transfer-learning-light https://data gen.tech/guides/computer-vision/resnet-50/

[3] K. Kamble, A. Bhansali, P. Satalgaonkar and S. Alagundgi, "Counterfeit Currency Detection using Deep Convolutional Neural Network," *2019 IEEE Pune Section International Conference (PuneCon)*, Pune, India, 2019, pp. 1-4, doi: 10.1109/PuneCon46936.2019.9105683.

[4] Ruheena, Bibi Fathima, Harshitha B M, Navyashree K, Dr. Shantharam Nayak, 2022, Fake Currency Detection using Deep Learning Technique, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) RTCSIT – 2022 (Volume 10 – Issue 12),

[5] B*, S*, B*, U*, & PH*, A. (3024). Fake Currency Detection Using Deep Learning. International Research Journal of Modernization in Engineering Technology and Science, 05(03), 3024. DOI: https://www.doi.org/10.56726/IRJMETS34730

[6] IJRASET. (2023). Detection of Fake Currency using Machine Learning. International Journal for Research in Applied Science & Engineering Technology, 11(IV), 4415.[6 www.ijraset.com

[7] A.A. Mandankandy, K.E. Kannammal, Fake currency detection: a survey. Gedrag en Organisatie 33(4), 622–638 (2020).

[8] A.Singh, K. Bhoyar, A. Pandey, P. Mankani, A. Tekriwal, Detection of fake currency using image processing. Int. J. Eng. Res. Technol. (IJERT) 8(12) (2019).

[9] G. Navya Krishna, G. Sai Pooja, B. Naga Sri Ram, V. Yamini Radha, P. Rajarajeswari, Recognition of fake currency note using convolutional neural networks. Int. J. Innov. Technol. Exploring Eng. 8(5), 58–63 (2019).

[10] K.D. Sudha, P. Kilaru, M.S.R. Chetty, Currency note verification and denomination recognition on Indian currency system. Int. J. Recent Technol. Eng. 7(6S) (2019). ISSN: 2277-3878

[11] M. Laavanya, V. Vijayaraghavan, Real time fake currency note detection using deep learning. Int. J. Eng. Adv. Technol. (IJEAT) 9(1S5) (2019). ISSN: 2249-8958.

[12] T. Kumar, T. Subhash, D. Regan, Fake currency recognition system for Indian notes

using image processing techniques (2019).

[13] A.Upadhyaya, V. Shokeen, G. Srivastava, Analysis of counterfeit currency detection techniques for classification model (2018).

[14] T. Agasti, G. Burand, P. Wade, P. Chitra, Fake currency detection using image processing, ICSET-2017. IOP Conf. Ser. Mater. Sci. Eng. 263, 052047 (2017).

Appendix: A- Packages, Tools used & Working Process

Python Programming language

Python is a high-level Interpreter based programming language used especially for general-purpose programming. Python features a dynamic type of system and supports automatic memory management.

It supports multiple programming paradigms, including object-oriented, functional and Procedural and also has its a large and comprehensive standard library. Python is of two versions. They are Python 2 and Python 3.

This project uses the latest version of Python, i.e., Python 3. This python language uses different types of memory management techniques such as reference counting and a cycle-detecting garbage collector for memory management. One of its features is late binding (dynamic name resolution), which binds method and variable names during program execution. Python's offers a design that supports some of things that are used for functional programming in the Lisp tradition. It has vast usage of functions for faster results such as filter, map, split, list comprehensions, dictionaries, sets and expressions. The standard library of python language has two modules like itertools and functools that implement functional tools taken from Standard machine learning.

Libraries

NumPy

NumPy is the basic package for scientific calculations and computations used along with Python. NumPy was created in 2005 by Travis Oliphant. It is open source so can be used freely. NumPy stands for Numerical Python. And it is used for working with arrays and mathematical computations.

Using NumPy in Python gives you much more functional behavior comparable to MATLAB because they both are interpreted, and they both allows the users to quickly write fast programs as far as most of the operations work on arrays, matrices instead of scalars. Numpy is a library consisting of array objects and a collection of those routines for processing those arrays.

NumPy has also functions that mostly works upon linear algebra, Fourier transform, arrays and matrices. In general scenario the working of NumPy in the code involves searching, join, split, reshaping etc. operations using NumPy.

The syntax for importing the NumPy package is → import NumPy as np indicates NumPy is imported alias np.

Matplotlib

Matplotlib is an indispensable library within the Python ecosystem, primarily employed for crafting a diverse array of visualizations ranging from static to interactive and even animated graphics. In the context of our Fake Currency Detection system project, Matplotlib plays a pivotal role in visually representing key aspects of our data, model training, evaluation, and result analysis. Below, we outline how Matplotlib is utilized within our project:

Data Exploration and Analysis: Before delving into model training, we employ Matplotlib to conduct thorough exploration and analysis of our dataset. Through various plot types such as histograms, scatter plots, and box plots, Matplotlib aids in comprehensively understanding the distribution of features, identifying potential patterns, and pinpointing outliers. This initial exploration phase enables us to make informed decisions regarding data preprocessing strategies.

Model Training and Evaluation: Throughout the iterative process of training our machine learning models, Matplotlib proves instrumental in visualizing critical metrics such as training and validation loss curves, as well as accuracy trends across epochs. By visualizing these metrics, we gain insights into the model's performance dynamics, enabling us to mitigate issues such as overfitting or underfitting and fine-tune model parameters effectively.

Confusion Matrix and Classification Reports: Post-training, we leverage Matplotlib to generate intuitive visualizations such as confusion matrices and classification reports, facilitating a comprehensive evaluation of the model's performance on our test dataset. These visualizations provide detailed metrics including precision, recall, and F1-score for each class, aiding in the interpretation of model behavior and efficacy in distinguishing between genuine and fake currency notes.

Visualizing Predictions: Matplotlib enables us to visually compare predicted labels against ground truth labels for a subset of our test dataset. This comparative analysis offers valuable insights into the model's predictive capabilities, allowing us to identify areas of strength as well as potential weaknesses that require further investigation and refinement.

Feature Importance: In scenarios where feature importance analysis is pertinent, Matplotlib assists us in crafting informative visualizations such as bar plots or heatmaps to highlight the relative importance of different features in predicting fake currency. These visualizations aid in identifying the most discriminative features and refining our feature selection strategies accordingly.

Keras: Simplifying Model Development

Keras served as our primary framework for building and training deep learning models. Its high-level API streamlined the process of defining neural network architectures, specifying loss functions, and configuring optimization algorithms. Leveraging Keras, we effortlessly constructed intricate neural networks tailored to our Fake Currency Detection task.

Keras Applications: Harnessing Pre-Trained Models

By incorporating Keras Applications, we capitalized on pre-trained deep learning models such as ResNet50. These pre-trained models, trained on extensive datasets like ImageNet, provided a robust foundation for our task. Leveraging transfer learning, we fine-tuned these models to detect fake currency with remarkable accuracy, even with limited data.

Keras Preprocessing: Preparing Image Data

Keras Preprocessing was instrumental in preparing our image data for model training. With its suite of utilities, we effortlessly loaded images from disk, resized them to the desired dimensions, and applied critical data augmentation techniques. This preprocessing step ensured that our model received properly formatted input data, enhancing its performance and robustness.

TensorFlow: Powering Deep Learning

TensorFlow, as a foundational deep learning framework, provided the backbone for our project. Its comprehensive ecosystem of tools and libraries facilitated every stage of our project, from model construction to deployment. We harnessed TensorFlow's high-level APIs like Keras for model development and its lower-level APIs for advanced customization, ensuring maximum flexibility and control.

Model Checkpoint: Ensuring Model Integrity

Model Checkpoint emerged as a critical safeguard for our trained models. By periodically saving the model's weights during training, we ensured its integrity and resilience against unforeseen interruptions. This ensured that we could seamlessly resume training from the best-performing checkpoint, preserving our progress and optimizing training efficiency.

Early Stopping: Preventing Overfitting

Early Stopping proved indispensable for preventing overfitting during model training. By monitoring specific metrics and halting training when performance ceased to improve, we safeguarded our model against memorizing training data and losing generalization capability. Early Stopping optimized training efficiency and ensured that our model achieved optimal performance without overfitting.

SGD (Stochastic Gradient Descent): Optimizing Model Training

SGD, as a cornerstone optimizer in TensorFlow, played a pivotal role in optimizing model training. Its stochastic gradient descent algorithm enabled us to efficiently train deep learning models by iteratively updating model parameters based on mini-batches of data. By customizing learning rates and momentum, we fine-tuned model training dynamics, achieving faster convergence and improved performance.

Tools Used Jupyter Notebooks:

Jupyter Project is spin-off project from the I-Python project, which is initially provided an interface only for the Python language. The name Jupyter is derived from the combination of Julia, Python, and R.

A Jupyter Notebook is basically a JSON file with a number of annotations. There are mainly three parts of the Notebook as follows.

i. **Metadata:** A data dictionary of definitions used to set-up and display the notebook; it is also be said as data about data.
ii. **Notebook format:** the Version numbers of the software used to create the notebook. The version number is used for backward compatibility**.**
iii. **List of cells:** there are three different types of different cells listed beside — markdown (display), code (to excite), and output.

Appendix: B

Sample Source Code with Execution

```python
checkpoint = ModelCheckpoint("Final_model.keras", monitor='val_acc', verbose=1, save_best_only=True, save_weights_only=False, mode='auto')
early = EarlyStopping(monitor='val_acc', min_delta=0, patience=40, verbose=1, mode="max")

finetune_model.compile(loss="categorical_crossentropy", optimizer=SGD(learning_rate=0.000001, momentum=0.9), metrics=['accuracy'])

finetune_model.fit(x=train_generator, epochs=num_epochs, validation_data=validation_generator, callbacks=[checkpoint, early])

# Evaluate the model on the test set
evaluation = finetune_model.evaluate(validation_generator)

# Save the model weights
finetune_model.save_weights("Final_4.weights.h5")
```

Python

```python
#testing the model
img=image.load_img(r"Data\Train\Real\1 (75).jpg" ,target_size=(300,300)) #The path of the testing image,the pic taken from the phone should come here

plt.imshow(img)
plt.show()
img=np.asarray(img)
img=np.expand_dims(img,axis=0)
finetune_model.load_weights("Final_4.weights.h5")

output=finetune_model.predict(img) #predicting the image using model created
print(output)

predicted_class_index = np.argmax(output)

# Map the index to the corresponding class label
predicted_class = class_list[predicted_class_index]

# Print the predicted class
print("Predicted class:", predicted_class)


# if(output[0][0]>output[0][1]):
#     print("Real")
# else:
#     print("Fake")
```